

This document is published in:

Advanced Concepts for Intelligent Vision Systems: 14th International Conference, ACIVS 2012. Proceedings (Lecture Notes in Computer Science, vol. 7517), pp. 491-502

DOI: [10.1007/978-3-642-33140-4_43](https://doi.org/10.1007/978-3-642-33140-4_43)

© 2012 Springer-Verlag Berlin Heidelberg

Estimation and Prediction of the Vehicle's Motion Based on Visual Odometry and Kalman Filter

Basam Musleh¹, David Martin¹, Arturo de la Escalera¹,
Domingo Miguel Guinea¹, and Maria Carmen Garcia-Alegre²

¹ Intelligent Systems Lab, University Carlos III, Leganes, Madrid 28911, Spain
bmusleh,dmmartin,escalera,dguinea@ing.uc3m.es

² Center for Automation and Robotics (CAR), Spanish Council for Scientific
Research (CSIC), 28500 Arganda del Rey, Madrid
maria@iai.csic.es

Abstract. The movement of the vehicle is an useful information for different applications, such as driver assistant systems or autonomous vehicles. This information can be known by different methods, for instance, by using a GPS or by means of the visual odometry. However, there are some situations where both methods do not work correctly. For example, there are areas in urban environments where the signal of the GPS is not available, as tunnels or streets with high buildings. On the other hand, the algorithms of computer vision are affected by outdoor environments, and the main source of difficulties is the variation in the lighting conditions. A method to estimate and predict the movement of the vehicle based on visual odometry and Kalman filter is explained in this paper. The Kalman filter allows both filtering and prediction of vehicle motion, using the results from the visual odometry estimation.

Keywords: Stereo vision, Visual odometry, Kalman filter.

1 Introduction

Various applications of Intelligent Transportation Systems (ITS), such as Advanced Driver Assistance Systems (ADAS) or autonomous vehicles need to have information about the movement of the vehicle. This information is usually supplied by a GPS, but there are some areas where the signal is not available in urban environments. This is because the signal is affected by high buildings or tunnels. Another option in order to know the movement of the vehicle is using the visual odometry. On the other hand, the visual odometry is affected by other kind of problems, mainly the light conditions make errors or do not allow to obtain the information about the vehicle's motion. A method to estimate the movement of the vehicle is presented in this paper, based on the visual odometry and whose results are filtered by using Kalman filter. Furthermore, if the method of visual odometry does not supply any estimation in a period of time, the Kalman filter can be used in order to predict the trajectory of the vehicle.

The visual odometry is one of the most active field in computer vision. Different solutions have made use of monocular cameras [18], stereo systems [4] and omnidirectional cameras [15]. Monocular sensors have the problem of scalar factor, but its implementation is easier than stereo systems which present a complex calibration step. Stereo systems achieve the most accurate results in long distances, because 3D information is available, although they present great uncertainty in depth estimation [3]. Several methods have been presented, working in the disparity space in order to solve the uncertainty [2]. Finally, omnidirectional cameras allow to track the feature points along more frames than other sensors [16]. On the other hand, the distortion makes difficult the feature matching. The presented method is based on a stereo vision system [12]. The visual odometry estimation is normally performed by means of detecting and tracking feature points between consecutive frames [14]. This visual odometry estimation makes use of the dense disparity map [17] to detect the road in front of the vehicle. Once the road has been detected, it is possible to only use the feature points that belong to the road, avoiding feature points of obstacles which can be a source of outliers if the obstacles are moving [20]. Another interesting result of using only feature points that belong to the road is the unnecessary search of feature points in the whole image, thus the developed algorithm only needs to process a lower third of the image.

The information of the road profile [8] is used to obtain the world coordinates of the road feature points. Besides the road profile, it is only necessary to know the position of the feature points on the left image in order to obtain the world coordinates, in contrast to most of the visual odometry algorithms, which need to perform a matching between the images of the stereo pair in order to obtain the disparity for each feature point. Moreover, the used feature points are close to the vehicle, reducing the uncertainty in depth estimation. Kalman filtering [7] has been applied to many situations in engineering, such as radio communication signals or applications to navigation. The filtering approach of the algorithm is applied in this work to raw data in order to smooth undesirable fluctuations in visual odometry. An acquisition of raw data from a GPS is synchronized with the capture of images in order to compare the results of the visual odometry with the GPS. The GPS is based on a Novatel receiver [5] that has been configured to work with Satellite-Based Augmentation Systems (SBAS) for sub-meter positioning. The solution has been calculated specifically with the European Geo-Stationary Navigation System (EGNOS), which is a type of geo-stationary satellite system that improves the accuracy of the basic GPS signals. Accuracy is enhanced through the use of wide area corrections for GPS satellite orbits and ionospheric errors. EGNOS consists of a series of reference stations, master stations, ground uplink stations and geostationary satellites (GEOs).

2 Obstacles and Road Detection

Two equal and parallel cameras can be used to obtain the depth (Z) for a point $P = (X, Y, Z)$ in world's coordinates by using (1), where the projections of point

P over the image planes are (u_L, v_L) and (u_R, v_r) for the left camera and the right one respectively.

$$Z = \frac{f \cdot B}{u_L - u_R} = \frac{f \cdot B}{d} \quad (1)$$

Where d is the disparity, f is the focal length and B is the baseline between both cameras. The disparity map is obtained by using the rectified images supplied by the stereo system, being the disparity (d) represented in the disparity map for every pixel of the image. Once the disparity map has been created, the u-v disparity [6] can be obtained. The v-disparity [8] expresses the histogram over the disparity values for every image row (v coordinate), whereas the u-disparity does the same, but for every column (u coordinate).

The dense disparity map and the u-v disparity are developed in order to detect obstacles in front of the vehicle. The method for detecting obstacles and the free space in front of the vehicle has been presented in [11]. This method obtains, as a result, two different dense disparity maps. The first one is the obstacle map (Fig.1(c)) and the second one, the free map (Fig.1(d)). The obstacle map is a disparity map where only the obstacles appear, whereas the free map is the opposite to the obstacle map, where only the empty space ahead of the vehicle appears. The obstacle detection has to detect every obstacle which blocks the movement of the vehicle, but it must not detect as an obstacle some possible elements which do not avoid the movement, for example a speed bump. Although the free map usually corresponds to the road, theoretically, is the empty whole space. This information can be used by a system of detection and localization of obstacles, as shown in [10].

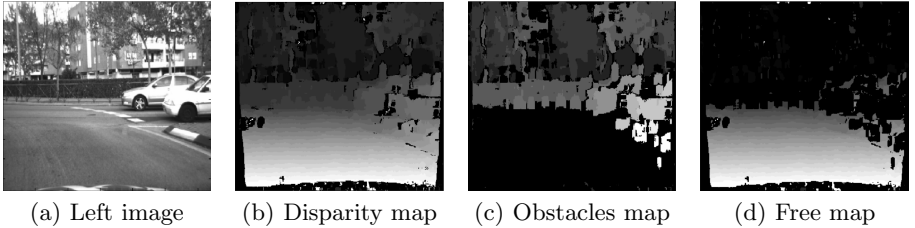


Fig. 1. Example of the obstacles and road detection

2.1 Estimation and Use of the Road Profile

As commented before, the depth (Z) for a point P can be calculated by means of (1), where the depth (Z) is a function of the disparity d , but the range of the disparity is low. This range is a function of the extrinsic and intrinsic parameters of the cameras, (for the results presented in this work, the range of disparity is $d = \{0, 25\}$). Thus, the resolution of (1) is sparse. The road profile [8] can be used in order to increase this resolution of the depth. The road appears as an oblique line in the v-disparity (road profile), which can be expressed as a straight line

(2) if a flat ground assumption is performed. Where v is the vertical coordinate of the image and b is the theoretical value of the horizon of the stereo system. Furthermore, once the road profile has been estimated, it is possible to know the pitch α between the stereo rig and the road [8] for each frame, by means of the (3) where C_v is the vertical coordinate of the optical center.

$$v = m \cdot d + b \quad (2)$$

$$\alpha = \arctan \frac{b - C_v}{f} \quad (3)$$

The road profile (2) shows the relationship between the vertical coordinate of the image v and the disparity d , then it is possible to obtain a new expression (4) merging the stereo equation (1) and the road profile (2), where now the depth (Z) is a function of the vertical coordinate of the image v . This new expression to calculate the depth Z is only for points belonging to the road. Moreover, in order to calculate the depth to the vehicle instead of the depth to the camera, it is necessary to use the information about the pitch α (3). Once the depth (Z) for a point belonging to the road has been calculated, it is possible to obtain its world coordinate X , which is a function of Z as (5) shows. Where C_u is the horizontal coordinate of the optical center.

$$Z = \frac{m \cdot f \cdot B}{v - b} \cdot \cos \alpha \quad (4)$$

$$X = \frac{Z \cdot (u - C_u)}{f} = \frac{m \cdot B \cdot (u - C_u)}{v - b} \quad (5)$$

It is important to note that estimating the road profile v-disparity which has been generated by using the disparity map, can be a difficult task in urban environments, so the detection of the oblique line corresponding to the road profile is difficult. An extended explanation of the solution to this problem is presented in [11]. For this reason, it is better to use the free map in order to generate the v-disparity because the obstacles are removed from the v-disparity, so it is easier to estimate the road profile.

3 Raw Visual Odometry Estimation

The visual odometry estimation is based on tracking feature points between consecutive frames of the left camera. In opposition to the usual methods of visual ego motion estimation or visual odometry [12], this method does not have to match up the feature points between both images of the stereo system to locate the points in world coordinates, because the method only uses the points belonging to the road. This points can be located on world coordinates by using (4) and (5) with the coordinates of points on the left image. An implementation of the Scale-Invariant Transform Feature (SIFT) detector and descriptor [9] developed for the MATLAB environment [19] is used in order to detect the feature points of the images.

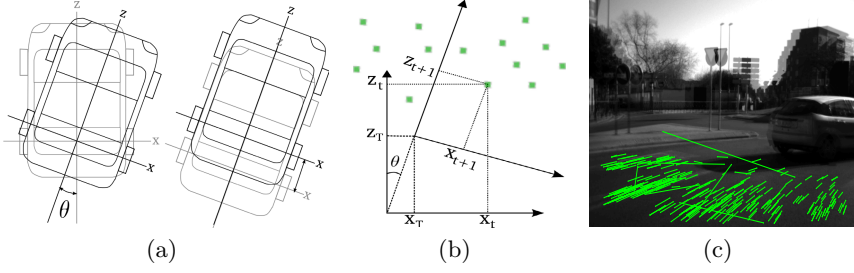


Fig. 2. (a) Schema of the movement of the vehicle. On the left, rotation stage. On the right, translation stage. (b) Schema of the movement of the cartesian coordinate system between consecutive frames. (c) Example of feature points detecting and matching between consecutive frames.

The chosen model for the kinematic motion of the vehicle is the Ackerman's steering model [1] [13]. In order to simplify the visual odometry estimation, it is necessary to make some assumptions: firstly, the movement of the vehicle between two consecutive frames can be divided into two stages, whose velocity is constant in each one (Fig.2(a)): a rotation around the center of the motion of the rear axle and a forward translation after the rotation. The second assumption is that there is no slip in any direction.

3.1 Estimation of the Visual Odometry between Consecutive Frames

The vehicle odometry estimation between two consecutive frames (t and $t+1$) is performed in two steps. Firstly, the feature points have to be detected on the two left images of the stereo pair, in the instants t and $t+1$. Then, a correspondence between the two sets of feature points (t and $t+1$) is done in order to know the displacement of the feature points within the images. As commented before, the method only makes use of points of the road which are situated at the bottom of the image. For this reason, it is possible to detect only the feature points at the bottom of the image, determining what feature points belong to the road by means of the free map. Besides the reduction of the computing time, using only the closest points to the vehicle, the flat ground assumption is improved. Once the image coordinates of the feature points have been obtained, it is possible to calculate the world coordinates $\{X, Z\}$ of these feature points in the instants t and $t+1$ by means of (4) and (5).

Secondly, the estimation of the vehicle movement between two consecutive frames (t and $t+1$) can be calculated by using the different locations in the instant t and $t+1$ of the points detected in the previous step. As Fig.2(b) shows, the rotation angle θ of the vehicle can be calculated by means of (6), where Z_T and X_T represent the translation after the rotation. Besides, it is possible to express Z_T and X_T as a function of θ and the locations of a road point, in the instant t and $t+1$ by using (7). On the other hand, the angle θ

is the only unknown variable of the expression in equation (8). It is possible to obtain firstly θ by solving the second order equation (9) and then Z_T and X_T by using (7).

$$\theta = \arctan\left(\frac{X_T}{Z_T}\right) \Rightarrow \tan \theta = \frac{\sin \theta}{\cos \theta} = \frac{X_T}{Z_T} \quad (6)$$

$$\begin{bmatrix} X_t \\ Z_t \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_{t+1} \\ Z_{t+1} \end{bmatrix} + \begin{bmatrix} X_T \\ Z_T \end{bmatrix} \quad (7)$$

$$\frac{X_T}{Z_T} = \frac{\sin \theta}{\cos \theta} = \frac{X_t - X_{t+1} \cos \theta - Z_{t+1} \sin \theta}{Z_t - Z_{t+1} \cos \theta + X_{t+1} \sin \theta} \quad (8)$$

$$(X_t^2 + Z_t^2) \sin^2 \theta + (2 \cdot X_{t+1} \cdot Z_t) \sin \theta + (X_{t+1}^2 - X_t^2) = 0 \quad (9)$$

In this way, a set of solutions $\{\theta, Z_T, X_T\}$ for the visual odometry estimation is obtained, where a solution $\{\theta, Z_T, X_T\}_k$ has been calculated for each pair of points $\{X_t, Z_t\}_k$ and $\{X_{t+1}, Z_{t+1}\}_k$. Different methods can be used in order to choose a unique solution $\{\theta, Z_T, X_T\}$ from the set of solutions, as a result of the visual odometry between the two consecutive frames. The algorithm uses a solution from the set by means of the median because it is robust to possible outliers.

4 Description of the Kalman Filter Implementation

The visual odometry estimation uses Kalman algorithm to solve filtering and prediction problem. The filtering approach of the algorithm is applied to raw data in order to smooth out undesirable fluctuations of visual odometry variables $\{\theta, Z_T, X_T\}$. Moreover, the Kalman algorithm is used if it is necessary to predict the evolution of the former variables in case of raw data loss.

4.1 Model Using in Kalman Filter

The process model is implemented by a linear time-varying (LTV) model in discrete time. That is, the process is described by a linear system. In this work, the system is a vehicle driving along a road at constant velocity. This linear system is a process, which can be described by the following two equations:

$$x_{t+1} = A_t \cdot x_t + w_t \quad (10)$$

$$y_t = B \cdot x_t + z_t \quad (11)$$

Where, t is the time index, x is the state of the system and y is the measured output. The variable w is the process noise and z is the measurement noise. The matrix A is the state transition matrix and B is the measurement matrix, which are obtained to model a simple vehicle moving with constant velocity.

Then, the state vector x consists of the vehicle location $p = [X, Z, \theta]$ and velocity v : $x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$, and the linear system equations are:

$$x_{t+1} = \begin{bmatrix} 1 & T_t \\ 0 & 1 \end{bmatrix} \cdot x_t + w_t \quad (12)$$

$$y_t = [1 \ 0] \cdot x_t + z_t \quad (13)$$

The Kalman filter combines the measurements from the system (variables for the visual odometry estimation $\{\theta, Z_T, X_T\}$), with the information provided by the motion model in order to obtain an optimal estimate of the system state. In this work, the measurement noise covariance matrix has been selected as the square of standard deviation of measurement (such as, 1^2 if we estimate Z_T), and the process noise covariance matrix uses process noise variance of 0.01.

4.2 Utilities of the Kalman Filter

The Kalman algorithm has been applied for filtering and prediction of visual odometry variables. Both utilities of the Kalman algorithm are displayed in Fig. 3, where three graphs represent the same example of a curve trajectory of the vehicle. The example shows only one of the three variables corresponding to odometry estimation, that is the angle θ . The example of the angle sequence starts at 20 second when the curve trajectory appears. The red circles of the graphs are visual odometry angles and blue circles are obtained solution with Kalman algorithm.

The first solution is the graph of Fig. 3(a), which presents the angle without fluctuations of the real data. The result allows to follow a smooth curve trajectory and eliminate outliers. In the graph can be observed that there are close measurements of 0.1 seconds where the algorithm smoothes correctly the curve trajectory. However, the most useful result appears when there is a gap of more than 0.1 seconds and an outlier is obtained by the visual odometry estimation, this outlier is usually caused by the shutter of the camera. In this special case, the Kalman filter obtains good results again as can be observed in the graph, smoothing the curve trajectory of the vehicle.

The second utility is prediction, where the Kalman algorithm is used to predict a solution when visual odometry cannot be calculated. Then, it is necessary to predict the evolution of the variables. We continue using the same curve trajectory as example and it is presented in (b) and (c) graph of Fig. 3. The experiment consists of requesting to Kalman algorithm an prediction of 100 measurements of odometry angle with a time interval of 100 milliseconds. The entire interval of 10 seconds is requested in two cases, first case when the vehicle is in the middle of the curve trajectory (Fig. 3(b)), and second case when the vehicle is reaching the end of the curve trajectory (Fig. 3(c)). Both prediction results are according to curve trajectory. In the right graph (Fig. 3(b)), the predicted angle presents an overshoot and finishes the estimated curve trajectory at 35 second. The third graph (Fig. 3(c)) shows the vehicle finishing again the estimated curve trajectory at 40 second and displays after a small overshoot. Both estimations reach a steady-state angle of 0 degrees.

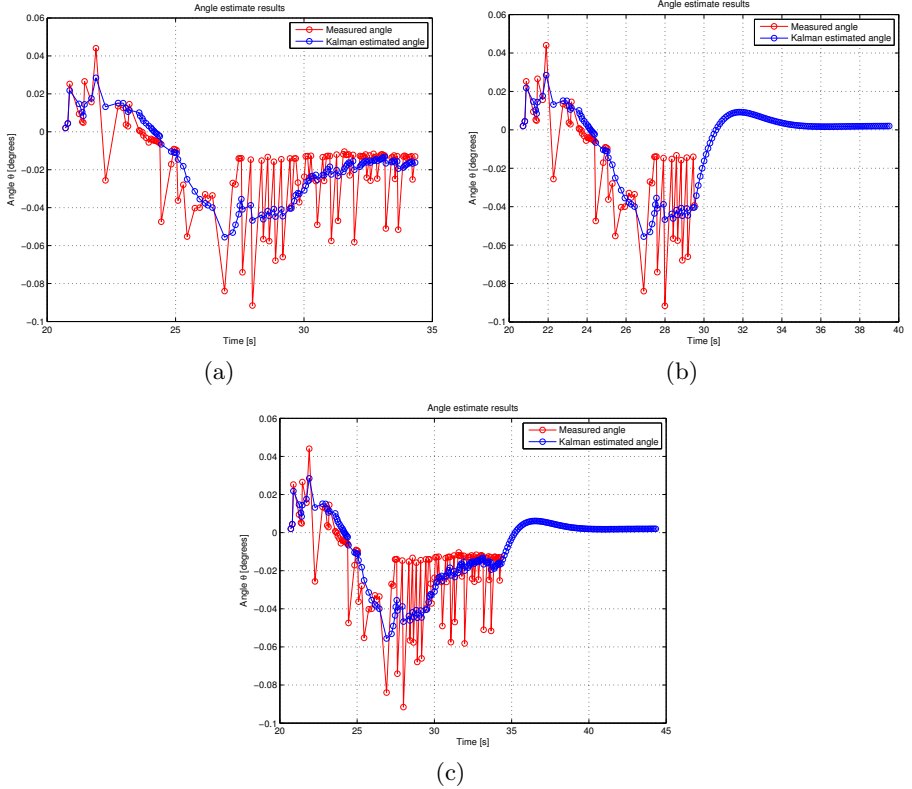


Fig. 3. Example of filter (a) and prediction ((b) and (c)) of θ raw data

5 Evaluation

Several tests have been performed in urban environments in order to evaluate the goodness of the algorithm of the visual odometry estimation. The results of a sequence, where the vehicle performs a closed loop in a urban environment, are presented in this section. The sequence has 1 minute and 38 seconds` duration and it is composed of 982 stereo images. Two different methods are used in order to evaluate the degree of accuracy of the visual odometry estimation. Firstly, it is possible to compare the difference between the initial and final location (position and orientation) of the vehicle. Due to the vehicle performs a closed loop, this two locations should be the same. Secondly, aerial imagery can be also used to overlay the resulting trajectory of the visual odometry estimation, and GPS raw data.

5.1 Results of the Visual Odometry Estimation

The first evaluation result is based on a comparison between the median and the mean, that has been performed in order to evaluate the robustness of the median

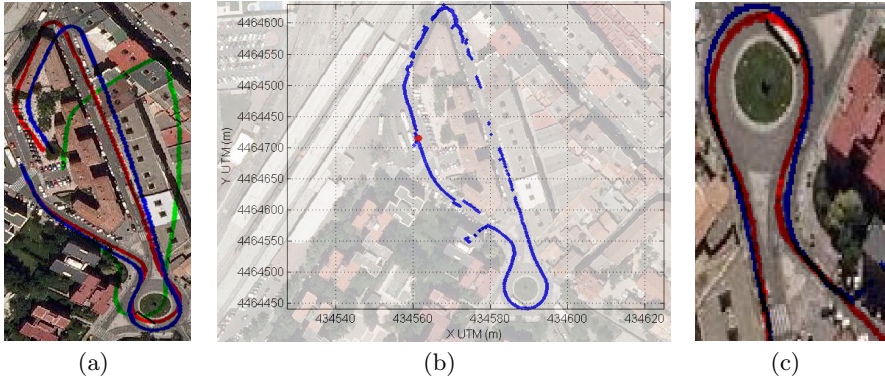


Fig. 4. Comparison between the different trajectories of the visual odometry estimation and GPS overlay in an aerial imagery. (a) Trajectory using obstacles in blue, mean in green and final visual odometry estimation (median) in red. (b) Raw data of the GPS overlay over an aerial imagery (c) Detail for comparison between the visual odometry estimation and the GPS raw data.

when it is used to choose the final solution $\{\theta, Z_T, X_T\}$. Fig. 4(a) presents the resulting trajectory of the visual odometry using both methods: the trajectory calculated by the mean appears in green and the median in red. The trajectory is deformed due to use the mean, in comparison with the obtained one with the median. The effect of not to use the feature points from the obstacles has been also studied. Fig. 4(a) shows the obtained trajectory using the feature points from the obstacles (blue colour). The fact that using feature points from the obstacles ahead of the vehicle produce a 50% position error higher than the visual odometry estimation without obstacles, whereas the rotation error is similar. Regarding to the trajectory, the deviation is more prominent in the entire trajectory as Fig. 4(a) shows.

Following, the raw data resulting from the visual odometry estimation are filtered in order to smooth fluctuations, Fig. 5 shows the results of filtering each variable $\{\theta, Z_T, X_T\}$. It is possible to observe the improvement of the smoothness in the three variables. The Fig. 5(a) displays the visual odometry estimated angle in red colour and Kalman filtered angle in blue colour. That is, Kalman algorithm estimates next angle using former angles of the time series. The results of the filter is appropriate in linear and curve trajectories, as can be observed in Fig.5(a), which first displays a linear trajectory followed by a curve to the left and so on. It is interesting to compare Fig.5(a) and Fig. 4(a) to observe the overall trajectory of the experiment, for example, when the vehicle is in the roundabout and how the Kalman filter smoothes continuously the curve trajectory to the left. The Fig.5(b) displays the X position and the behaviour of the filter is accurate obtaining again a smooth trajectory. It can be observed again the equivalence between Fig.5(b) and Fig. 4(a). The third graph is the Z position where the Kalman algorithm filters fluctuations of approximately 0.8 meters. The algorithm smoothes efficiently the Z variable.

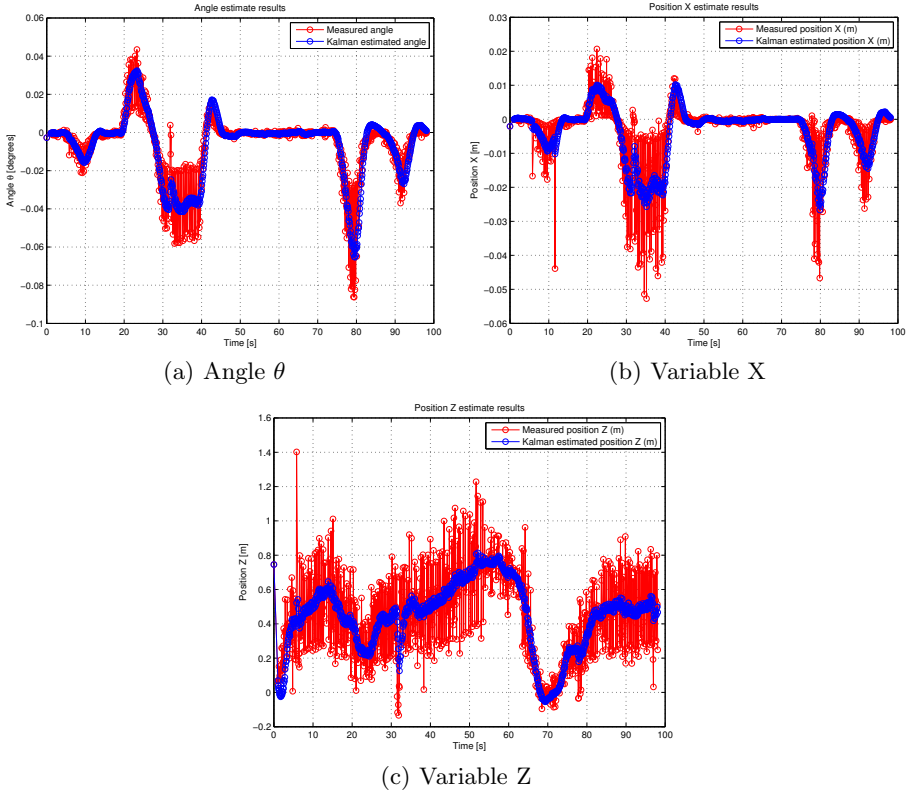


Fig. 5. Results of the raw data filtering from visual odometry estimation

5.2 Comparison between Visual Odometry Results and GPS

The second evaluation result is based on GPS. The GPS raw data of the followed trajectory by the vehicle is displayed in Fig 4(b). Raw data is shown in Universal Transverse Mercator (UTM) geographic coordinate system to compare visual odometry estimation data and GPS raw data in meters. The trajectory start, which is followed by the vehicle, is marked with a red dot in Fig. 4(b), and each blue dot stands for a GPS point. The accuracy of overall trajectory is less than one meter when GPS + EGNOS is active, and if the receiver uses single point L1 solution the accuracy is 1.5 meters. Moreover, if the receiver is working in single point L1 solution and the number of GPS satellites is insufficient to calculate an optimum solution, then the accuracy of the solution can be increased more than 1,5 meters, resulting in a lateral displacement of the blue dots close to the real trajectory of the vehicle. The trajectory shows some gaps, where the receiver cannot compute the solution caused by the loss of GPS satellites. The loss of satellite signals is caused by obstructions from close buildings in the right and left side of the road. The comparison between visual odometry estimation results and GPS raw data can be observed in Fig. 4(c). The visual odometry trajectory is

indicated in red colour and GPS raw data in blue colour. This comparison is the tool that allows to establish the performance of the visual odometry estimation. Considering the accuracy of the GPS receiver, explained in former paragraph, the results establish that visual odometry has better performance than GPS raw data. It is possible to observe wrong data at the bottom-right of the Fig.4(c) due to the receiver cannot compute the solution caused by the loss of GPS satellites. The loss of satellite signals is caused by a close building in the right of the road. The red trajectory matches exactly with the cars in the aerial image, as can be observed at the beginning of the roundabout where appears a car that is waiting for entering in the roundabout. A second example is in the middle of the roundabout, where a bus is performing a curve trajectory.

6 Conclusions

The 2D visual odometry estimation has been explained and applied in urban environments. The advantages of the smart algorithm have been shown comparing GPS raw data with visual odometry results. The robust visual odometry estimation ensures safe trajectory in case of GPS raw data loss caused by buildings, trees, tunnels, among other solid elements around the vehicle. The GPS drawbacks have been solved with proposed algorithm. The visual estimation has reached the accuracy for curve and linear trajectories of the vehicle, avoiding outliers from dynamic obstacles. The results display a position error of 3.2% and a rotation error of 2.6% in a close loop, which accomplishes the goal of the estimation algorithm. In addition, the use of the road profile and free map information allow to search feature points only in the lower third of the left images, and reduce the uncertainty in depth estimation because these points are close to the vehicle. Kalman filter has been used as a great asset for smoothing out undesirable fluctuations of visual odometry variables, and if necessary the Kalman algorithm can predict the variables of the vehicle's movement in case of raw data loss.

Acknowledgments. This work was also supported by Spanish Government through the CICYT projects FEDORA (Grant TRA2010-20255-C03-01), Driver Distraction Detector System (Grant TRA2011-29454-C03-02) and by CAM through the projects SEGVAUTO-II.

References

1. Borenstein, J., Everett, H., Feng, L.: Where am i? sensors and methods for mobile robot positioning. University of Michigan 119, 120 (1996)
2. Demirdjian, D., Darrell, T.: Motion estimation from disparity images. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 213–218. IEEE (2001)
3. Hernández, A., Nieto, J., Vidal Calleja, T., Nebot, E., et al.: Large scale visual odometry using stereo vision (2011)

4. Howard, A.: Real-time stereo visual odometry for autonomous ground vehicles. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 3946–3952. IEEE (2008)
5. NovAtel Inc., Calgary (2012), <http://www.novatel.com>
6. Hu, Z., Lamosa, F., Uchimura, K.: A complete uv-disparity study for stereovision based 3d driving environment analysis. In: Fifth International Conference on 3-D Digital Imaging and Modeling, 3DIM 2005, pp. 204–211. IEEE (2005)
7. Kalman, R.: A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(Series D), 35–45 (1960)
8. Labayrade, R., Aubert, D., Tarel, J.: Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In: Intelligent Vehicle Symposium, vol. 2, pp. 646–651. IEEE (2002)
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
10. Musleh, B., Escalera, A., Armingol, J.: Real-time pedestrian recognition in urban environments. In: Advanced Microsystems for Automotive Applications, pp. 139–147 (2011)
11. Musleh, B., de la Escalera, A., Armingol, J.: U-v disparity analysis in urban environments. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2011, Part II. LNCS, vol. 6928, pp. 426–432. Springer, Heidelberg (2012)
12. Nistér, D., Naroditsky, O., Bergen, J.: Computer Vision and Pattern Recognition. In: Proceedings of the 2004 IEEE Computer Society Conference on CVPR 2004 , vol. 1, p–652. IEEE (2004)
13. Nourani-Vatani, N., Roberts, J., Srinivasan, M.: Practical visual odometry for car-like vehicles. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3551–3557. IEEE (2009)
14. Parra, I., Sotelo, M., Llorca, D., Ocana, M.: Robust visual odometry for vehicle localization in urban environments. *Robotica* 28(3), 441–452 (2010)
15. Scaramuzza, D., Fraundorfer, F., Siegwart, R.: Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 4293–4299. IEEE (2009)
16. Scaramuzza, D., Siegwart, R.: Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics* 24(5), 1015–1026 (2008)
17. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* 47(1), 7–42 (2002)
18. Stein, G., Mano, O., Shashua, A.: A robust method for computing vehicle ego-motion. In: Proceedings of the IEEE Intelligent Vehicles Symposium, IV 2000, pp. 362–368. IEEE (2000)
19. Vedaldi, A.: An open implementation of the SIFT detector and descriptor. Tech. Rep. 070012, UCLA CSD (2007)
20. Wangsiripitak, S., Murray, D.: Avoiding moving outliers in visual slam by tracking moving objects. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 375–380. IEEE (2009)